



SERVICE CENTER IMPLEMENTATION TEAM (SCIT)

DATA MANAGEMENT TOOLS: REQUIREMENTS/STRATEGY

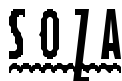
- DBSM/Mobile Computing
 - Data Warehousing & Data Marts
 - GIS, Complex Data, Message Queuing, Middleware

Submitted to:

U.S. Department of Agriculture,
OCIO/CCE

Submitted by:

SOZA
& Company, Ltd.
8550 Arlington Blvd.
Fairfax, Virginia 22031



USDA
Farm Service Agency
Natural Conservation
Resources Service
Rural Development



June 15, 1999

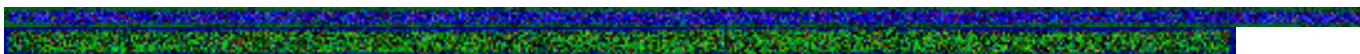


TABLE OF CONTENTS

1	Section 1—Introduction	1
1.1	Background.....	1
1.2	Objective.....	2
2	Section 2—Assumptions	3
2.1	End-User Delivery Environment Assumptions.....	3
2.2	Applications Environment.....	3
2.2.1	Decision Support Systems	4
2.2.2	Warehouses and Marts	5
2.2.3	Online Transaction Processing Using Local Data Stores.....	5
2.2.4	Message Queuing	6
2.2.5	GIS	6
2.2.6	Locally Unique Data	6
2.3	Data Environment.....	6
2.3.1	National Level Applications	7
2.3.2	State and Regional Applications	7
2.3.3	Service Center Level Applications	8
2.4	Assumptions for Future Service Center Applications	8
3	Section 3 - Approach.....	9
3.1	Assumptions	9
3.2	Roles and Responsibilities	10
4	Section 4—Evaluation Strategy	11
4.1	Categories of Applications.....	11
4.1.1	Decision Support Systems	11
4.1.2	Warehouses and Marts	11
4.1.3	Highly Complex Documents.....	11
4.2	Online Transaction Processing Using Local Data Stores.....	11
4.3	Message Queuing	11
4.4	Locally Unique Data	12
4.5	GIS	12
4.6	Evaluation Steps.....	12
4.6.1	Step 1—Determine Appropriate Criteria for Categories of Applications	12
4.6.2	Step 2—Weighting Evaluation Criteria Groups	12
4.6.3	Step 3—Evaluation Guidelines	12
5	Section 5—Overall DBMS Evaluation Criteria.....	13
5.1	Performance - Requirements for DBMS and local store databases	13
5.1.1	Metrics: Database Loads.....	13
5.1.2	Metrics: Restructure Unload and Reload	13
5.1.3	Metrics: Replication Server to Server and Server to Mainframe.....	13
5.1.4	Metrics: Multi-user with high volume of simultaneous Queries and Updates	13
5.1.5	Metrics: Recovery Time Subsequent to Hardware/Software Crash.....	13
5.1.6	Transaction Processing Council Benchmarks	13
5.2	Operation(s) and Controls	13
5.2.1	Locking	13
5.2.1.1	Page Level Locks	13
5.2.1.2	Table Level Locks.....	14
5.2.1.3	Database Level Locks	14
5.2.2	Dirty Read Mode.....	14
5.2.3	Support for Loosely Coupled Systems.....	14

5.2.4	Array Interface	14
5.2.5	Asynchronous I/O	14
5.2.6	Shared Log Files	14
5.2.7	Non-Blocking Queries	15
5.2.8	Clustered Indexes	15
5.2.9	Cost- and Statistics-Based Optimizer	15
5.2.10	Optimizer	15
5.2.11	Stored Procedures	15
5.2.12	Stored Functions in Database	16
5.2.13	Performance Monitoring Tools	16
5.3	Integrity - Requirements for DBMS	16
5.3.1	Adherence to Industry Standards	16
5.3.2	Declarative Integrity Model	16
5.3.3	Cascading Delete	16
5.3.4	Null Support	16
5.3.5	Triggers	16
5.3.6	Event Alerters	17
5.4	Database Administration	17
5.4.1	Portable	17
5.4.2	Automatic Database Recovery	17
5.4.3	Multiplexed Log Files	17
5.4.4	Database Mirroring	17
5.4.5	Online Database Backup	18
5.4.6	Online Recovery	18
5.4.7	DBA Utilities	18
5.4.8	Remote Maintenance of Database	18
5.5	Distributed RDBMS	18
5.5.1	Distributed Join	18
5.5.2	Synchronous Table Replication	18
5.5.3	Asynchronous Table Replication	19
5.5.4	Connections to Other Databases	19
5.5.5	Programmatic Two-Phase Commit	19
5.5.6	Remote Procedure Calls (RPCs)	19
5.5.7	Use of Triggers and Predefined Procedures	19
5.5.8	SQL-Based Database Gateway	19
5.5.9	ODBC Support	19
5.6	Database Security	19
5.6.1	O/S Security Integration	20
5.6.2	User Group Privileges and Roles	20
5.7	Languages and Tools	20
5.7.1	SQL Procedural Language	20
5.7.2	Support for Extended Data Types	20
5.7.3	Union Operator	20
5.7.4	Select for Update	20
5.7.5	Outer Join Operator	20
5.7.6	Dynamic SQL	20
5.7.7	Static SQL	20
5.7.8	Transaction Savepoints	21
5.7.9	Aliases and Synonyms	21
5.7.10	Graphics Tools	21
5.7.11	Internal E-Mail System Integration	21

5.7.12	National Language Support	21
5.7.13	Precompiler Support	21
5.7.14	Web Support	21
5.7.15	XML Support	21
5.8	Centralized Meta Data Repository Inter-operability (CMDR)	22
5.9	Enterprise 7x24 Operations and System Administration	22
5.10	Middleware	22
5.10.1	Middleware Embedded Support	22
5.11	Cost	22
5.11.1	Cost To Procure	23
5.11.2	Cost To Maintain/Upgrades	23
5.11.3	Cost for Training	23
5.11.4	Cost To Implement	23
5.12	Support – Requirements for DBMS	23
5.12.1	Technical Support	23
5.12.2	Vendor Future – Requirements for DBMS	24
5.12.3	Current Market Assessment (Financial)	24
5.13	Interoperability – Requirements for DBMS	24
5.13.1	Access to Open Protocols	24
5.14	New Technology Criteria – Requirements for DBMS	24
5.14.1	Web Server	24
5.14.2	Query & Reporting	24
5.14.3	Meta Data Management	25
5.15	Section 7 – Mobile Computing (MC) Requirements	26
5.15.1	DBMS Support of Mobile Windows NT Workstation Applications	26
5.15.2	MC: Enterprise Impact	26
5.15.3	MC: Systems Integration	26
5.15.4	MC: Performance	26
5.15.5	MC: Specialized Requirements	26
5.15.6	MC: Replication Support for Disconnected Computing	26
5.15.7	MC: Performance, Reliability, and Scalability	27
5.15.8	MC: Replication	27
5.15.9	MC: Ease of Use	27
5.15.10	MC: Heterogeneous Support	27
5.15.11	MC: Immediate Update	27
5.15.12	MC: Internet Support	27
5.15.13	MC: Merge Replication	27
5.15.14	MC: Multi-Site Update	28
5.15.15	MC: Scalability	28
5.15.16	MC: Desktop and Mobile Systems	28
5.15.17	MC: Automatic Tuning	28
5.15.18	MC: Code Compatibility	28
5.15.19	MC: Embedded Version	28
5.15.20	MC: Integration With Microsoft Access	28
5.15.21	MC: Mobile Clients and Replication	28
5.15.22	MC: Universal Data Access	28
5.15.23	MC: Integration with Microsoft Office 2000	28
5.15.23.1	MC: Connectivity	29
5.15.23.2	MC: Developer Tools	29
5.16	CASE Tools Interoperability	30
5.17	Questions for Vendors – Requirement for DBMS	30

6	Section 6—Data Warehouse (DW) and Data Mart Requirements	31
6.1	DW: Architecture	31
6.2	DW: Specialized Schemas	32
6.3	DW: Operational Environment	32
6.4	DW: Operates within the Common Computing Environment (CCE)	32
6.5	DW: Define an Enterprise Architecture that evolves and scales	32
6.6	DW: ODBC Support	32
6.7	DW: Security	32
6.8	DW: UNIX Compatibility	32
6.9	DW: Compatible Data Recovery	33
6.10	DW: Multi-tasking	33
6.11	DW: Infrastructure (Telecommunications) Compatibility	33
6.12	DW: Business Objectives	33
6.13	DW: Architecture	33
6.13.1	DW: Multiple Staging Areas with Synchronization	33
6.13.2	DW: Data Administration	34
6.13.3	DW: Extraction, Transformation, and Translation	34
6.14	DW: Data Delivery	34
6.15	DW: High Speed and Volume Loader with Disparate Sources	34
6.16	DW: OLAP Tool and Reporting Extensions	34
6.17	DW: Loader Customizations	35
6.18	DW: Data Warehouse to GIS Extensibility	35
6.19	DW: Multi-tasking and Parallel Processing	35
6.20	DW: Transaction Processing Council Benchmarks	36
6.21	DW: Specialized Requirements	36
6.21.1	DW: Very Large Databases (VLDB)	36
6.21.2	DW: Symmetrical Multiprocessing (SMP)	36
6.21.3	DW: Splitting a table(s) across many physical devices	36
6.21.4	DW: Load/unload, query, update, index building within multiple processors	36
6.21.5	DW: Asynchronous I/O	36
6.21.6	DW: Clustered Indexes	36
6.21.7	DW: Query Optimization	36
6.21.8	DW: Terabyte commercial installations	37
6.21.9	DW: SQL92 for data definition and data manipulation	37
6.21.10	DW: SQL implementation (null or not)	37
6.21.11	DW: Outer Joins	37
6.21.12	DW: SQL92 data types (business information)	37
6.21.13	DW: Spatial Data	37
6.21.14	DW: Extensible for new types of data and standards	37
6.21.14.1	DW: Variety of options/tools for performance optimization	37
6.21.15	DW: Variety of Backups	38
6.21.16	DW: Single warehouse for spatial data and future use	38
6.21.17	DW: Backup Warehouses	38
6.21.18	DW: Efficient loading of the Data Warehouse into Data Marts	38
6.21.19	DW: Parameters supporting the migration of data from legacy systems	38
6.21.20	DW: Large Volume Data Transfers	38
6.22	DW: Decision Support Systems	38
6.22.1	DW: Highly Complex Data/Documents	39
6.22.2	DW: Functionality and Extensibility	39
6.22.3	DW: WEB Enabled Document Transfer and Reports	39
6.22.4	DW: Application Development Tools (Inclusion / Interfaces)	39

6.23	DW: Query Performance	40
6.23.1	DW: Parallel index scan and parallel execution plans?	40
6.23.2	DW: Query rewrite facility	40
6.23.3	DW: Thread Performance.....	40
7	Section 7 - Message Queuing and Middleware Support	41
7.1	Image Management	41
7.2	Vector Data Management	41
8	Section 8 – Geographical Information Systems (GIS)	42
8.1	Management of Imagery (DOQ).....	42
8.2	Ability To Manage Compressed Data	42
8.3	Vector Graphics.....	42
8.3.1	Boolean Algebra.....	42
8.3.2	Multiple User-Specified Coordinate System	42
8.3.3	Ability to Tile GIS.....	43
8.3.4	DBMS to maintain vector data history.....	43
8.4	Support for GIS Spatial Query Operators	43
9	Section 9 – Complex Data.....	45
10	Section 10—Conclusion	46

LIST OF FIGURES

FIGURE 2-1. THE VISION IS ACHIEVED BY INFORMATION TECHNOLOGY; MODIFIED FUTURE VISION..	4
FIGURE 2-2. SERVICE CENTER DATA MANAGEMENT MODEL	7
FIGURE 3-1. THE SERVICE CENTER ENTERPRISE ARCHITECTURE.....	9

1 Section 1—Introduction

1.1 Background

The Data Management Tools Technical Working Group (TWG) develops the requirements for selecting enterprise data management tools to support Service Center Business Process Reengineering (BPR) applications. These tools include the data management systems to manage tabular data, geospatial data, and complex data types such as images, video, and voice.

The architectures for these tools span a wide range of environments, including mainframe, server, and desktop. Collectively, these tools provide the data management environment for the following:

- Client/server applications
- Web applications
- Server information
- Desktop office automation tools
- Geographic Information System (GIS) tools.

This tool suite includes major data management engines and any associated middleware required for a complete data management solution.

The TWG works with Common Computing Environment (CCE) and local area network (LAN)/wide area network (WAN)/Voice to establish an integrated suite of scalable, interoperable tools that manage data for reengineered business applications to support the three Farm Agencies (FSA, RD, & NRCS). Over time, legacy systems will be reengineered or ported to this new environment. Current proposals call for the migration of legacy systems by the end of 2002. The goal would be to have a manageable data environment for the Database Management System (DBMS), GIS, and document management. A comprehensive solution reduces the number of systems that must be supported, which reduces training, maintenance, and system administration costs. A single solution may not be feasible, but every effort should be made to minimize the number of configurations. The requirements for the data management tools will be derived as a collective requirement from the initial BPR projects.

The data management tools will be bundled with and procured as part of CCE. The implementation strategy for CCE has not been finalized. However, the current proposal calls for purchasing the Service Center Network Servers; some minimal implementation of desktops and laptops; and the components to implement Administrative Convergence Applications (e.g., PeopleSoft and the HR application) in the first part of FY99 in Phase 1. In the second phase (FY00), the GIS data servers and the public access servers will be purchased.

To work within the CCE, the requirements for the major data management tools should be determined before the GIS and public access servers are purchased. A major dependency exists between the GIS tools and the DBMS. At a minimum, this interoperability must be defined after the requirements from the BPRs have been more fully developed and as BPR improvements are tested and piloted in the Integration Lab. A more detailed and final requirement for the architecture to support national roll out should be specified before deploying the CCE. If the current CCE implementation proposal stands, this would mean that the major requirements for data management tools would need to be known by early calendar year 1999.

1.2 Objective

The Service Center Implementation Team (SCIT) Data Management Tools TWG develops the requirements for data management tools, which include the DBMS, server-based GIS tools, and other tools as identified. This TWG also develops documentation to support the procurement of enterprise-wide data management tools. The scope of the enterprise includes the delivery of applications and data at the national, regional, state, and Service Center levels in support of the Service Center BPR initiative.

2 Section 2—Assumptions

This section outlines the future environment that the data management tools must support. They must support collecting and disseminating of information in a multi-tiered environment. The Data Management Tools TWG recognizes the requirement of Service Centers to collect and manage local information that is not shared outside their internal organizations.

An initial assumption was that this group would work in conjunction with the Data Architecture TWG to develop a baseline assessment of the current data infrastructure that supports the various business areas. The Data Architecture TWG did not meet in time to provide the Data Management Tools TWG with the necessary data architecture information. Assumptions relative to the three Farm Agencies as an enterprise were made. The following sections outline the assumptions made by the Data Management Tools TWG in the delivery, applications, and data environments, and as well as about the hardware architecture.

2.1 End-User Delivery Environment Assumptions

- Network operating system¹—Windows NT
- Cabling²—Category 5, 10base-T
- Internal network protocol—TCP/IP
- Service Center access to the WAN—via a 28.8k baud modem dialed by a router
- Web browsers—Microsoft and Netscape
- NT printer service
- NT file service
- NT Server and Workstation 4.0
- NT Authentication
- MS Office 97 with Access and Outlook 98
- Laptop with interrupted connectivity to the network
- Continuous WAN connections in offices such as state offices
- Sufficient local online storage space available
- Backup technology available at the Service Centers.

Network performance projections, staff skills, and business communication capabilities will determine the locations of centralized and decentralized data servers.

2.2 Applications Environment

Data management tools must support the varied applications within the three Farm Agencies. **Figure 2-1** lists the types of applications that the tools must support.

¹ Common Computing Environment, Service Center Technical Architecture Study, February 1998

² LAN/WAN/Voice Technical Environment Study, December 1996

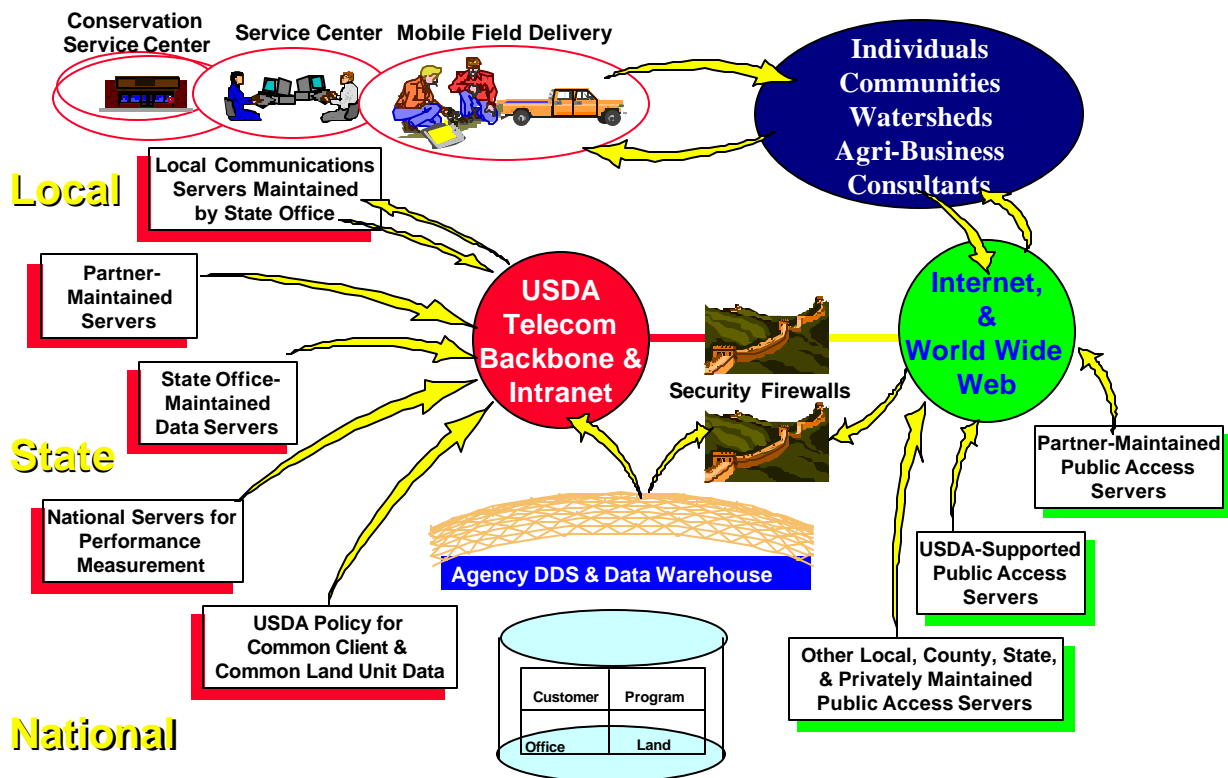


Figure 2-1. The Vision Is Achieved by Information Technology; Modified Future Vision³

These applications are based on the CCE Baseline Technical Architecture Inventory of February 1997⁴, which was used to develop and validate the group's assumption that the applications currently fielded would influence the choice of future data management tools. Appendix A contains an extract from the inventory. The following sections briefly discuss the types of applications used by the Service Centers.

2.2.1 Decision Support Systems

At the agency level, Decision Support Systems (DSSs) provide input into the decision-making process of an organization. Traditionally, these are supported by extracts from the unstructured normalized data warehouse repository to support a users based on a business area specification of the dimensions of content and deployed as data marts that focus on a single business functional area such as personnel or recruiting. Decision support systems are a combination of analysis tools and views of data in either the data warehouse, or more frequently, the data mart.

OLAP (OnLine Analytical Processing) Decision support software allows the user to quickly analyze information that has been summarized into multidimensional views and hierarchies. For example, OLAP tools are used to perform trend analysis on financial information. They can enable users to drill down into masses of transaction history in order to isolate expenditures that are over estimates.

Traditional OLAP products, also known as multidimensional OLAP, or MOLAP, requires a multidimensional "cube" to be extracted ahead of time. User queries on these types of databases are extremely fast, because most of the consolidation has already been done.

³ NRCS Future Vision, May 1997, Modified

⁴ Business Needs and Technology Alternative Evaluation Study, November 1997

A relational OLAP, or ROLAP, tool extracts data from a traditional relational database. Using complex SQL statements against relational tables, it is able to create the multidimensional views on the fly. ROLAP solutions are usually considered to be more scalable (with the size of the data limited only by the scalability of the DBMS that hosts it) and usually have a lower total cost of administration and ownership.

2.2.2 Warehouses and Marts

A Data Warehouse is a central repository in which enterprise-wide information is cleansed, integrated and organized to provide a foundation for critical business intelligence applications. Warehouses contain a nearly current state and history of the enterprise and are subject-oriented normalized time-variant repositories designed to hold large amounts of data; they are updated from an online transaction segment that maintain the current state of the enterprise.

Within the USDA, data from multiple source legacy systems will be transformed and cleansed in nearly real-time feeds the USDA Data Warehouse. In turn, the USDA Data Warehouse will serve as a staging area to normalize and integrate data from disparate operation systems before feeding sub-sets of that data into specialized data marts. It is in the consolidated Data Warehouse that new and meaningful relationships will be built out of previously isolated applications. The ability to relate geo-spatial with traditional business data in the warehouse should be particularly powerful. Data integrity will improve by standardizing on the metadata model of the warehouse. Total cost of ownership will improve through central administration of the data and the interfaces.

Although analysis will usually be performed against the data mart specific to a business process, reports may be run directly against the data warehousing. For instance, operational reports on historical, normalized, non-aggregated data will be supported at the warehouse level.

Data marts are collections of integrated subject-oriented denormalized time-variant structured databases designed to support the Decision Support function of a particular business unit, each unit of data is relevant to some moment in time; they are refreshed from updates to the warehouses using extract and transformation tools.

As a process and architecture, a data warehouse and data marts require robust planning to implement performance oriented data structure(s) and platforms fitting the separate needs of the warehouse repository and the mart repositories. The planning phases include selecting, converting, transforming, consolidating, integrating, cleansing, and mapping recent and historical data from multiple operational data sources to a target DBMS.⁵ A data warehouse should support the enterprise decision-making processes and provide the organization with intelligent business systems. Data warehouse-data mart architecture provides flexibility and extensibility to support the applications an enterprise knows it requires today and the applications that it will require in the future.

Within the USDA, several data mart projects are already underway. Today, these data marts are fed directly from operational systems. In the future, the preferred model will be to build data marts from specialized “partitions” of the Data Warehouse. Therefore, the data flow will be from online transaction systems to the data warehouse and then to data marts. Users will design the dimensions (content) of the mart from the universe of enterprise data, rather than from a single application.

2.2.3 Online Transaction Processing Using Local Data Stores

Numerous Service Center-based Online Transaction Processing (OLTP) applications process against local data stores, such as the Automated Claims System. This processing method enables the local office to

⁵ Handbook of Data Management 1998, B. Thuaisingham, Auerbach, 1998

establish, adjust, refer, transfer, collect, and control Commodity Credit Corporation (CCC) and Farm Service Agency (FSA) claims against producers on a Local Office Claim File.

2.2.4 Message Queuing

Message queuing, or store-and-forward messaging, is based on intermediate message storage. A queue is a data store of messages in process. The application sends the request to the messaging middleware, which places it in a queue that may reside on the client system, the final destination system, or another node in the network. Message queuing is asynchronous in a manner similar to a traditional paper mail system—the recipient need not be available when the message is sent. Database replication servers, e-mail systems, and most Electronic Data Interchange (EDI) systems are examples of applications where delivery of messages can be deferred.⁶

2.2.5 GIS

During next few years, U.S. Department of Agriculture (USDA) is investing about \$250 million in the acquisition, integration, and delivery of geospatial data. The data team recognized that the collection, storage, management, and delivery of GIS data have unique requirements for data management tools. The delivery of geospatial information will be made through GIS⁷.

2.2.6 Locally Unique Data

The implementation of BPR projects that empower Service Center employees is increasing. The Integrated Office Information project is developing applications that will generate locally stored information that will not to be shared outside the local office. CCE has identified MS Access and an SQL server as options that may support this function.

2.3 Data Environment

For the purposes of this document, the enterprise is defined as all the applications developed to support the three Farm Service Agencies. **Figure 2-2** indicates the data management team's vision of how the data layers relate and the emphasis placed on centrally managed data. The following sections discuss the assumptions made about each of those levels.

⁶ Gartner Group Strategic Analysis Report R-400-102, S. Varma, February 9, 1996

⁷ USDA Service Center Geographic Information System (GIS) Strategy, June 1998

The model emphasizes centralized management of data definition and decentralized data management of physical data.

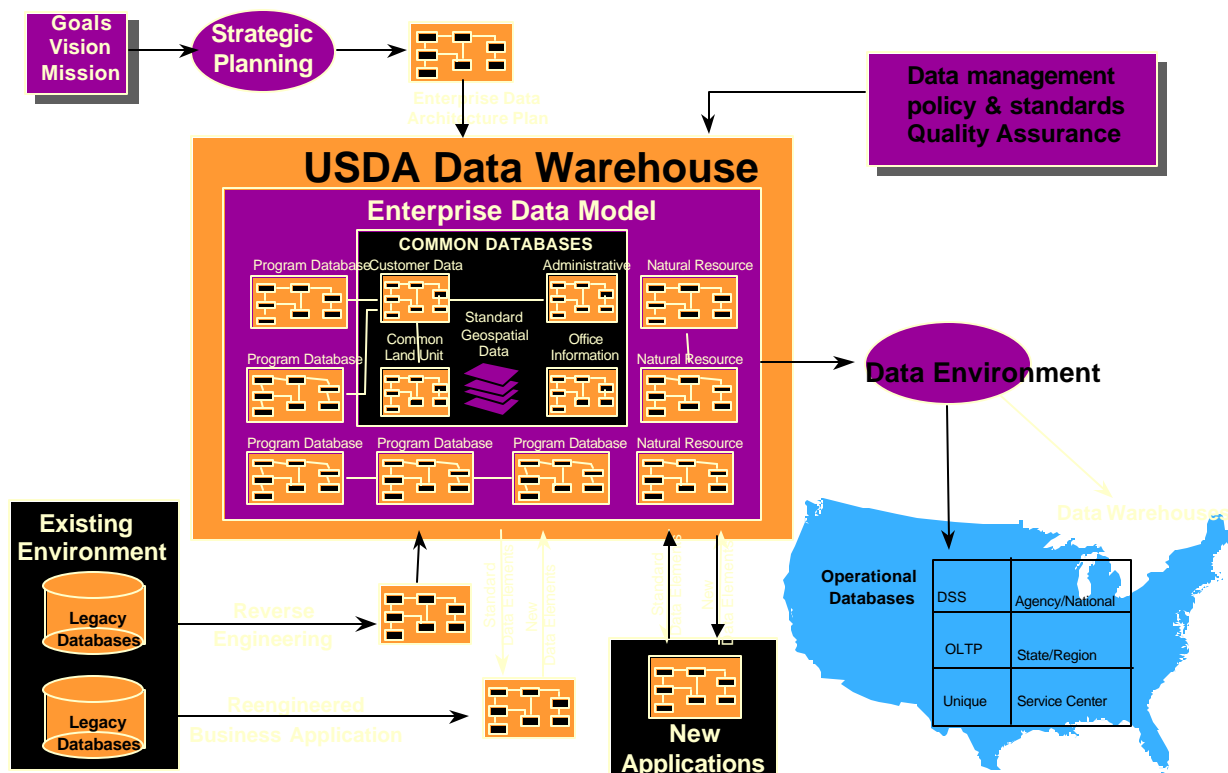


Figure 2-2. Service Center Data Management Model

2.3.1 National Level Applications

Several requirements support national applications. First, data may be generated and stored at the Service Center, then aggregated at the national level. This requires a data management tool that supports synchronization and replication across a number of tiers within the organization. Second, a centrally managed program, such as a commodity program, is hosted at the national level with no data stored at the local level.

Both types of applications will continue in the reengineered environment that the CCE and the data team will have to consider when they select DBMS tools to support the business requirements. The Consolidated Administrative Management System (CAMS) initiative illustrates a reengineered business process that implements a centrally managed database to support of human resources management for the three agencies.

2.3.2 State and Regional Applications

All three agencies have fielded applications targeted for the state and region. Rural Development traditionally fields their Rural Housing from a Service Center and their region. In a number of FSA applications, the information is collected locally and processed at the state office.

2.3.3 Service Center Level Applications

The SCIT BPR initiative addresses activities and functions conducted at the Service Center. A number of the FSA applications run on IBM System 36s at the Service Center level and transmit data to either state or agency level computers for processing. With the exception of NRCS's Field Office Computing System (FOCS), most applications are FSA farm loan programs that issue checks at the Service Center level.

2.4 Assumptions for Future Service Center Applications

The following assumptions concern Service Center database applications to be delivered over the next two years:

- The requirement to field Agency Level, two Tier OLTP applications and locally stored information will remain.
- The three Service Center partner agencies will have to manage a dramatic increase in geospatial data.
- Web-based delivery of applications will increase for all three agencies.
- Web-based applications will be generated using tools such as Cold Fusion and Net Dynamics.
- Applications will be developed using tools such as Visual Basic, Cool Gen, PowerBuilder, Infomaker, XML, C, and C++.

3 Section 3 - Approach

Figure 3-1 shows the overall approach used for developing the data management tools selection strategy.

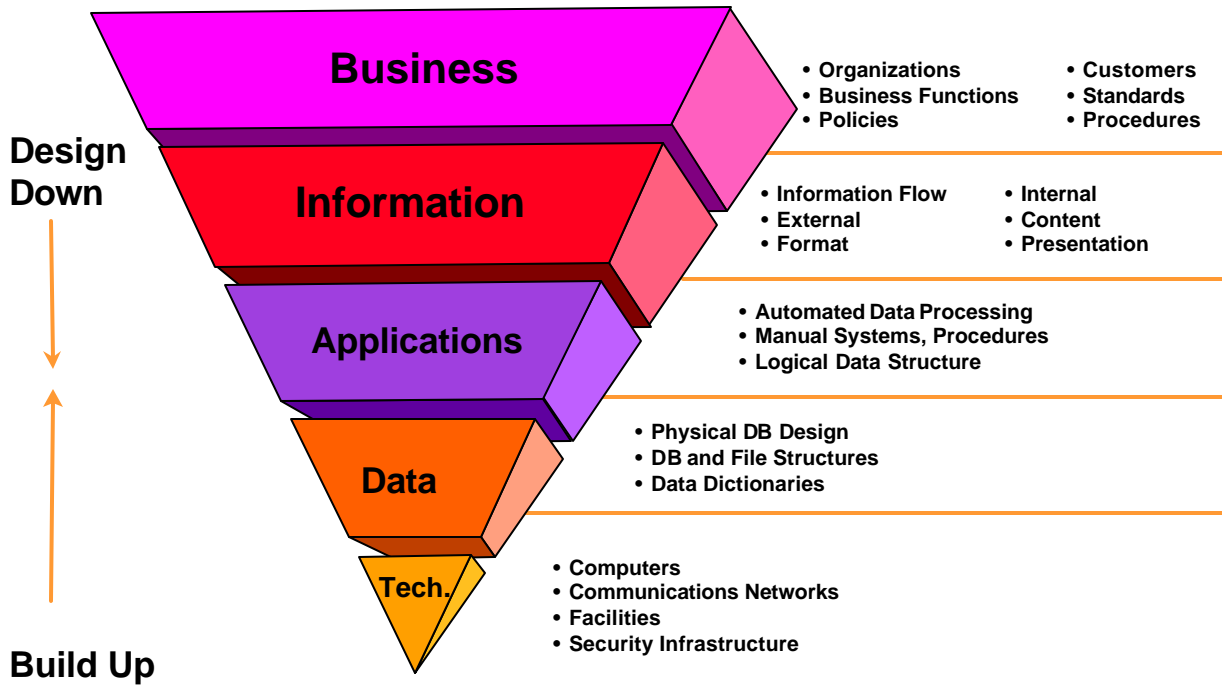


Figure 3-1. The Service Center Enterprise Architecture

3.1 Assumptions

To determine the delivery platform at the service centers, the Data Team reviewed the draft Service Center Operating Environment developed by the CCE. The delivery environment is outlined in Sections 2.3 and 2.4. The applications delivery assumptions account for the future applications that the data management tools would have to support.

The Data Team worked with representatives from CCE and LAN/WAN/Voice communities to define the roles and responsibilities of the data team in the evaluation process for the data management tools. After an extensive search for industry-accepted criteria, the team assembled more than 100 evaluation criteria for selecting data management tools. Appendix B contains this list, which was later narrowed to 65 differentiating criteria.

The assumption is that the criteria removed could be met by all vendors and would be non-differentiating. The list was supplemented with criteria for GIS and middleware. Section 4 outlines the evaluation strategy.

Insight into the data requirements for the current technical architecture was gained from the CCE baseline inventory. The Gartner Group analyzed market trends for the various data management tools as well as the future direction of the tool vendor's product line. Appendix C contains a synopsis of the Gartner

Group's research. The 1997-2002 USDA Strategic Plan provided insight into the overall mission goals and objectives of the three partner agencies.⁸

3.2 Roles and Responsibilities

Information from the BPR projects will help to define the data management tools requirements that support the various applications related to the project. The CCE team is responsible for purchasing the data management tools. The Data Management TWG will work with CCE and LAN/WAN/Voice teams to establish an evaluation and selection approach for data management tools. The Data Management TWG may participate in the selection of the DBMS for the enterprise.

⁸ USDA Strategic Plan 1997 - 2002

4 Section 4—Evaluation Strategy

The strategy outlined in this section will be used to evaluate data management tools that will support the three Farm Service Agencies' applications. The SCIT Data Management TWG met to define the evaluation categories of vendors' applications, as well as the criteria to be used during the evaluation process. Risks and hidden implementation requirements will be identified by testing known constraints.

4.1 Categories of Applications

The Service Center Data Management Tools TWG has grouped the applications that support the Farm Agency business functions into the following categories: Decision Support Systems, Warehouses and Marts, Highly Complex Documents, Online Transaction Processing Using Local Data Stores, Messaging and Queuing, Locally Unique Data, and GIS.

4.1.1 Decision Support Systems

DSSs, applications running at the agency level, provide input into the decision-making process of an organization. Traditionally, these are considered data marts that focus on a single functional area such as personnel.

4.1.2 Warehouses and Marts

Warehouses and marts are a collection of integrated subject-oriented databases designed to support the decision support system function, where each unit of data is relevant to some moment in time. A data warehouse is a process and an architecture that requires robust planning to implement a platform. The phases include selecting, converting, transforming, consolidating, integrating, cleansing, and mapping recent and historical data from multiple operational data sources to a target DBMS. A data warehouse should support the enterprise decision-making processes and provide the organization with intelligent business systems. A data warehouse architecture must be flexible enough to support the applications an enterprise knows it requires today and the applications that will be required in the future.

4.1.3 Highly Complex Documents

A number of SCIT initiatives deal with highly complex documents, such as the directives BPR, which will result in a single mechanism for obtaining program policy information across the three agencies. This type of application can store electronic documents in a database and access components within a document.

4.2 Online Transaction Processing Using Local Data Stores

Numerous Service Center-based OLTP applications process against local data stores, such as the Automated Claims System. This processing method enables the local office to establish, adjust, refer, transfer, collect, and control CCC and FSA claims against producers on a Local Office Claim File.

4.3 Message Queuing

Message queuing, or store-and-forward messaging, is based on intermediate message storage. A queue is a data store of messages in process. The application sends the request to the messaging middleware, which places it in a queue that may reside on the client system, the final destination system, or another node in the network. Message queuing is asynchronous in a manner similar to a traditional paper mail system—the recipient need not be available when the message is sent. Database replication servers, e-

mail systems, and most EDI systems are examples of applications where delivery of messages can be deferred.

4.4 Locally Unique Data

The implementation of BPR projects that empower Service Center employees is increasing. The Integrated Office Information project is developing applications that will generate locally stored information not to be shared outside the local office. CCE has identified MS Access and MS SQL Server as options that may provide this functionality.

4.5 GIS

Over the next few years, USDA is investing about \$250 million in the acquisition, integration, and delivery of geospatial data. The data team recognized that the collection, storage, management, and delivery of GIS data have unique requirements for data management tools. The delivery of geospatial information will be made through GIS.

4.6 Evaluation Steps

The following section describes the steps used to evaluate data management tools.

4.6.1 Step 1—Determine Appropriate Criteria for Categories of Applications

In this step, the evaluators will choose the appropriate criteria to be applied to each category of application. Vendor product lines can be evaluated against about 95 criteria, but not all of the criteria will apply to all categories of applications.

4.6.2 Step 2—Weighting Evaluation Criteria Groups

This step prioritizes the combinations of database management systems and protocols on their ease of integration, sharing of information systems, and support of operating systems. Evaluators will assign a percentage value for each criteria group. The overall distribution of weighted value per type of application cannot exceed 100 percent

4.6.3 Step 3—Evaluation Guidelines

In Step 3, the evaluators rate each tool vendor's product against the criteria established for that tool. Evaluators may leave cells blank if they do not have the background in the specific area being evaluated. A scale of : blank, 1-10, assesses the vendor's product against the criteria.

Blank—A blank indicates that the vendor's product does not have the specific functionality described in the evaluation criteria.

1-3—A range of 1 to 3 indicates whether the vendor plans to provide that specific functionality for the first time within the next 3- 6 months; or that the capability is marginally acceptable.

4-6—A range of 4 to 5 indicates that the vendor's product somewhat meets the criteria established.

7-9—A range of 6 to 8 indicates that the vendor's product meets to exceeds the criteria established.

10—A range of 9 to 10 indicates that the vendor's product line exceeds the criteria and that the vendor is recognized as an industry leader providing that specific functionality.

5 Section 5—Overall DBMS Evaluation Criteria

5.1 Performance - Requirements for DBMS and local store databases

This segment will identify those categories that will only be measurable by imperial benchmark evaluations for effectiveness. Actual tests and use of independent studies are recommended.

Transaction Performance Metrics:

The product shall be measured against the following performance categories to determine and rate their metrics:

5.1.1 Metrics: Database Loads

5.1.2 Metrics: Restructure Unload and Reload

5.1.3 Metrics: Replication Server to Server and Server to Mainframe

5.1.4 Metrics: Multi-user with high volume of simultaneous Queries and Updates

5.1.5 Metrics: Recovery Time Subsequent to Hardware/Software Crash

5.1.6 Transaction Processing Council Benchmarks

The product to qualify should have been benchmarked on the TPC-D (Operational transactions) tests. The scores should be viable information for speed and capacity analysis.

Have TPC build a matrix of transaction volume demand, database size, hardware specs, software, number of users and performance scores to create a map of preferences that will depend on the enterprise growth parameters.

5.2 Operation(s) and Controls

5.2.1 Locking

The smallest level of granularity in commercial databases is the row level. Next comes page level, table level, and database level. Typically, database pages are 2,048 bytes, so if each row contains 100 bytes of data, a user might squeeze 20 rows into a page. Row level locking means that the database can use a lock granularity of a single row. Therefore, multiple users (up to 20 in the example above) can simultaneously update different rows on the same page. Each user, when performing an operation on the row, locks only that row in question and does not interfere with other users in the same page. Arguments for row level locking are that it permits the highest degree of concurrency—users tend not to lock each other out. Arguments against include a claim that row level locking causes a lot of overhead.

5.2.1.1 Page Level Locks

When one user updates a row, the entire page is locked and other users are blocked from updating (sometimes reading too) rows in that page. Sometimes users get around page level blocking by forcing

what would be small rows to take up an entire page. This workaround effectively simulates row level locking for critical tables that are being updated frequently. The arguments for and against page level locking are mentioned above.

5.2.1.2 Table Level Locks

Locks can also occur for an entire table. This feature is useful for locking a table for batch updates, locking out users for maintenance, or generating reports.

5.2.1.3 Database Level Locks

The entire database should also be able to be locked with a single command.

5.2.2 Dirty Read Mode

The database has a read mode that scans the data as it exists on disk or in memory, regardless of whether it has been committed or not. In general, dirty reads are fast, but since the data are not necessarily valid, some applications cannot use them. This feature is helpful for generating reports where accuracy does not matter.

5.2.3 Support for Loosely Coupled Systems

Loosely coupled processors are usually independent computers connected by a fast communications bus that can share and coordinate resources. Databases that work in this environment can run simultaneously on several nodes in a cluster. This increases power since processing occurs at each node, and it reduces complexity since no two-phase commit or distributed database work is needed. Fault tolerance is lessened because if one of the machines stops running, the remaining nodes keep processing. Today, several UNIX machines, such as Pyramid, Sequent, and IBM RS/6000, offer clustering in addition to DEC. Performance concerns may exist if OLTP goes against the same data on more than one node since database blocks ping around the cluster if this happens too much. Oracle works this way on Massively Parallel machines like Ncube, KSR, and Maspar.

5.2.4 Array Interface

This feature reduces network traffic; for example, “send me 1,000 rows, thanks,” vs. “send a row, thanks, send a row, thanks” 1,000 times. This is an issue only for cursor-based processing; for streams-based processing, this is not an issue.

5.2.5 Asynchronous I/O

Most operating systems do provide asynchronous I/O, including MS, MVS, VM, and UNIX. For systems that do not provide asynchronous I/O, the need to wait for disk writes to complete may become a performance bottleneck. The server should be able to take advantage of asynchronous I/O where available and have a strategy to deal with the lack of asynchronous I/O on systems where it is not available.

5.2.6 Shared Log Files

Every change to the database structure is automatically written in a “piggybacked” manner to the redo file structure. Transactions are piggybacked in a way that one physical write can actually commit several transactions. This is a very efficient way to commit transactions because it requires only a sequential write to an O/S file. Multiple users can share the same redo file structure.

5.2.7 Non-Blocking Queries

For systems where multiple users are reading and writing to the same table at the same time, it may be critical that readers writing reports do not interfere with ongoing transaction processing. When readers need to obtain locks to get information, they are said to “block” writers since writers are blocked from getting the locks they need to perform their update. This feature is useful on a transaction processing system that runs reports against live data during operating hours.

5.2.8 Clustered Indexes

Clustered indexes can be defined as an index on a cluster key or a pre-sorted index cluster that greatly speeds data retrieval for ordered operations at the expense of insert and update speed. Oracle uses the first type; Sybase the second.

5.2.9 Cost- and Statistics-Based Optimizer

The database will gather and store comprehensive statistics about database structures. The statistics-based optimizer will choose the most efficient access paths to data based on the information gathered. An efficient system can estimate the statistics on a table based on a sample of the data. This is important when it takes an application 4 hours to analyze statistics for a 30-gigabyte table. Another feature of an efficient optimizer takes into account low/high data values and data distribution heuristics.

5.2.10 Optimizer

The optimizer scans the data to determine the minimum value in the table, the maximum value, and the average value. It also keeps track of the distribution of data within the rows.

The optimizer can estimate statistics by scanning a random sample of the rows in the table. This feature is very useful for collecting statistics on large tables where it is impractical to scan the whole table.

The optimizer can take into account real-time system load information in determining the optimal path for queries. Depending on the system load, the optimizer may change the path.

A classic example of this feature occurs when a local table with ten rows is being joined to a remote table with one million rows. The optimizer sends the local table to the remote node for the join rather than the other way around.

5.2.11 Stored Procedures

Stored procedures are bits of procedural code, grouped and stored in the DBMS engine. Stored procedures are usually stored in shared, compiled format. Stored procedures should be callable by applications, by other stored procedures, or by database triggers. There should also be full dependence tracking that will automatically recompile stored procedures when objects upon which they depend change. Stored procedures are efficient because they typically improve performance and reduce network traffic. They also encapsulate allowable operations on data (most systems let a user grant access to a stored procedure without granting access on the underlying tables). However, since there is no standard for stored procedures, any code a user writes that uses the procedures is non-standard and non-portable.

Stored procedures are cached in memory on the server. They do not need to be read in from the disk each time they are called. A special, Database Administrator (DBA) configurable area of memory holds the procedure cache, so it does not get flushed by users performing large queries. A single copy of the stored procedure can be used by multiple users. This saves memory and execution time.

Stored procedures can return an array or table of data. They are not limited to returning only a single row.

Stored procedures can return all available server datatypes, not just a limited subset.

5.2.12 Stored Functions in Database

The user can define stored function calls. These user-defined functions are useful for many reasons, such as having both procedures and functions in a 3GL language.

5.2.13 Performance Monitoring Tools

The vendor provides tools to monitor system performance and diagnose problems. The tools monitor individual SQL statements and overall system performance.

5.3 Integrity - Requirements for DBMS

5.3.1 Adherence to Industry Standards

Of the numerous RDBMS standards, the most useful is FIPS 127-1 since the Government tests the products to ensure their compliance. SQL2 and SQL3 standards exist or are under development, but no official test suites are being proctored. All are complex standards that define a common SQL dialect in detail. The advantage of this is that if users write FIPS 127-1 code, they will be able to run their applications against all adhering databases with little rewriting of code. For users to be able to port code to another DBMS without a rewrite, they must use only 100 percent ANSI standard statements and a database that has been certified by NIST.

5.3.2 Declarative Integrity Model

This model includes full support for declarative referential integrity, default values, and domain support. The ANSI-declarative approach greatly simplifies the process of providing database enforced integrity. It allows the programmer to define primary and foreign keys, default values, unique keys, and so on when creating database tables. The database engine automatically enforces these rules to protect system data. This model is superior to databases that make a user program referential and entity integrity using stored procedures and triggers. The Declarative Method is portable, standard, and low maintenance.

5.3.3 Cascading Delete

Cascading delete supports the cascade delete feature of the declarative referential integrity model. This feature deletes all the corresponding children if the user deletes the parent record. All of the products that support triggers can achieve this programmatically.

5.3.4 Null Support

The programmer can create a table and specify whether or not null values can exist in each column. The SQL implementation should provide a function that will determine whether the value in the column is null or not. This feature is useful when performing arithmetic operations and outer joins. The database should also correctly evaluate to false *null = anything*.

5.3.5 Triggers

Database triggers are procedural codes associated with tables and are fired implicitly when data are modified in their table. They are used for access control, data validation, referential integrity, synchronous table replication, and other uses.

Database triggers can automatically fire once, and only once, per SQL statement. These are useful for security; for example, when the trigger might check whether the date is a Sunday, and if it is, fail the SQL statement.

5.3.6 Event Alerters

Events can be defined in the database, which the database will watch for. When the event occurs, the database engine will take some pre-determined action. For example, when the inventory drops below a certain level in the inventory table, an event alerter notices, sends a message to the purchasing clerk, and automatically enters a purchase order for the product needed.

5.4 Database Administration

5.4.1 Portable

The server should run on many different types of hardware and operating systems. Three-tier portability includes installations on microcomputers, minicomputers, and mainframes. Some databases also run on super computers and massively paralleled computers. Hardware portability is an important feature in selecting the best performing hardware. Other aspects of portability include network protocol portability and graphical user interface portability.

5.4.2 Automatic Database Recovery

Database failures are usually grouped into several categories. Instance failure, occurs when the machine running the database server crashes, or software fails, or an uninformed operator damages the server, with no losses on the database disks. Media failure occurs when a disk containing database information fails.

Database recovery in case of instance failure should be performed automatically by simply restarting the database instance. The engine should roll back any transactions that were pending but not committed at the time of failure, and ensure the integrity of all committed transactions. The time to recover from instance failure should be configurable by the DBA.

Recovery from media failure should be automatic, semi-automatic, or manual. In all cases, recovery from media failure requires that a backup of the lost files at some point in time is available, along with all database redo files since that point in time.

5.4.3 Multiplexed Log Files

The database can maintain multiple transaction redo files on different disks and write to all of them at the same time. This provides added protection in case of a loss of the disk upon which the log files reside, which otherwise would render the database non-recoverable for up-to-the-minute transactions.

5.4.4 Database Mirroring

The RDBMS can perform software disk mirroring of the database files regardless of whether the mirroring is supported at the hardware or operating system level. Mirroring means keeping multiple copies of all database information to protect them from disk failure. All of the products can take

advantage of hardware and operating system disk mirroring, which are preferable to RDBMS software mirroring if they are available.

5.4.5 Online Database Backup

DBAs can backup of the entire database online while the database is up, all tables are online, and users are active. This should not require locking and should have a minimal effect on system performance, other than the I/Os required to perform the backup.

5.4.6 Online Recovery

The product support online recovery since this is critical for OLTP and 24-hour-a-day mission critical systems. Online recovery is the ability to recover subsets of the database while the rest of the database is up and online. The database does not need to be taken offline during recovery of a data file. Users who do not require access to failed areas of the database will remain unaware that failure occurs. Users who do attempt to access damaged files will receive an appropriate error message and will be able to access the file as soon as recovery is complete. Some databases may require that the entire database be taken offline for recovery from media failure.

5.4.7 DBA Utilities

The DBA utilities should be able to start and stop an engine, perform real-time monitoring of database use and performance, assist in backup and recovery of database logs and data, and provide for execution of SQL statements. Utilities may also help in managing users, user groups, application privileges, and more. The DBA utilities should also be able to run in client/server mode to facilitate centralized management.

5.4.8 Remote Maintenance of Database

Since there are over 2,400 remote Service Centers, Technical personnel must have the capability to log on to these remote sites to troubleshoot problems and provide database maintenance. Some areas of concern would include root privileges that would allow remote backups, unlocking threads, restarts, etc.

5.5 Distributed RDBMS

5.5.1 Distributed Join

The product should provide the capability of joining two tables that are on different machines into a select statement. Some systems can perform this transparently to the user and to the application program. The query is then exactly the same as if the tables resided on the same machine.

Example:

```
SELECT ENAME, DEPTNAME FROM EMP, DEPT  
WHERE EMP.DEPTNO=DEPT.DEPTNO;
```

Where the EMP table resides on the local server, and the DEPT table resides on a remote server.

5.5.2 Synchronous Table Replication

Table replication allows for the distribution of updates from a single master to one or more slave nodes. Synchronous replication implies that updates are propagated to slave nodes in real time, protected by a two-phase commit. A classic use for table replication occurs in banking transactions, when balances must be synchronized on all machines at all times. A drawback for synchronous replication occurs when one node fails and the transaction does not go through on any of the nodes.

5.5.3 Asynchronous Table Replication

Asynchronous replication differs in that updates are not propagated in real time. This is useful for information that does not always need to be perfectly in synch. Also, with asynchronous table replication, all nodes do not need to be alive for the initial transaction to complete. Updates get propagated throughout the network at programmer-defined intervals.

5.5.4 Connections to Other Databases

Gateways are available that facilitate the incorporation of data from foreign databases. Gateways are read/write, read only, procedural, or SQL based.

5.5.5 Programmatic Two-Phase Commit

Two-phase commit (TPC) is a mechanism for managing distributed update capabilities in a distributed database. Two-phase commit ensures that an update either completes or rolls back on all nodes in a transaction. Programming for this feature is very complex and becomes exponentially more difficult as the number of nodes in a transaction grows.

5.5.6 Remote Procedure Calls (RPCs)

Stored procedures may be called from remote nodes without restriction. Translation of character sets and data types automatically occurs. Stored procedures can perform remote updates or call other stored procedures on remote nodes. Any distributed transactions are automatically protected by TPC.

5.5.7 Use of Triggers and Predefined Procedures

The OLTP system will host a variety of new and updated applications. As the data entry point for USDA, the data integrity and security will be weighted higher for the OLTP DBMS. Data integrity (and business logic) may be enforced by DBMS 'triggers' and pre-defined procedures. The OLTP system should be evaluated for widest range of trigger and user-defined procedure options.

5.5.8 SQL-Based Database Gateway

Ad hoc SQL statements can be sent to foreign databases for processing. The application thinks that it is sending an SQL statement to the local server. Features that should be considered include SQL dialect translation, functionality augmentation, static versus dynamic SQL, and performance.

5.5.9 ODBC Support

The product should support (or has announced support for) all ODBC standards.

5.6 Database Security

5.6.1 O/S Security Integration

O/S security integration implies the ability to specify that database logins will be validated by the operating system. That is, each account maps one-to-one with operating system login identifiers. On IBM systems, database security is integrated with the host security management system. The O/S should not broadcast unencrypted passwords across the network.

5.6.2 User Group Privileges and Roles

The system supports simple user groups, like UNIX, where access rights can be grants to group names rather than individuals. Roles are collections of privileges like user groups. Roles differ from simple user groups in terms of additional functionality. Users may be granted several roles and can switch between roles in a session. Users can also enable and disable specific privileges within their role. Roles are additive in nature; thus a manager role can have the privileges of clerk plus additional rights.

5.7 Languages and Tools

5.7.1 SQL Procedural Language

Because native SQL does not work well for algorithmic processing, most vendors have devised procedural languages. These languages are used to write stored procedures and triggers, and often can be used as standalone programs.

5.7.2 Support for Extended Data Types

The DBMS should support extended data types such as documents and geospatial data.

5.7.3 Union Operator

Standard union combines multiple tables and eliminates duplicate rows. The ANSI standard also includes the UNION ALL operator, which does not eliminate table duplicates and is useful for horizontal partitioning of data.

5.7.4 Select for Update

Select for update is useful for obtaining locks on desired structures and can be used if repeatable read functionality is required. It can be used to test if a structure is already locked by another user, like the TSET instruction or executing a spinlock. It is required for ANSI compliance.

5.7.5 Outer Join Operator

Outer join lets users perform a join where they want all the rows from one table, even if no matching rows exist in the second table. This feature is very useful for modeling real world problems.

5.7.6 Dynamic SQL

Ad hoc SQL statements can be generated and sent to the database engine for processing. They do not need to be separately compiled and have plans generated before they can be executed.

5.7.7 Static SQL

SQL statements can be pre-parsed, compiled, and have plans generated. Thus, they execute faster since this does not have to be done at run time. This feature improves speed at the expense of flexibility and maintenance. This section explicitly does *not* include stored procedures and triggers, which typically are stored compiled.

5.7.8 Transaction Savepoints

Savepoints are markers used in transactions to increase their atomicity. They are coupled with the *rollback to savepoint* command. This prevents an entire multi-part transaction from being rolled back by the failure of one of the individual statements within it.

5.7.9 Aliases and Synonyms

Aliases and synonyms are used as substitute names for tables, views, and so on. These features hide the fact that a table is actually owned by a different schema. Aliases and synonyms are also used to implement location transparency for distributed databases:

5.7.10 Graphics Tools

The vendor provides tools that allow graphical applications to be built that use database data.

5.7.11 Internal E-Mail System Integration

Mail systems should be written specifically to run on top of and integrate with the database system. In general, this facilitates the combining of database data and reports into the mail system. Database events can trigger mail messages, and vice-versa.

5.7.12 National Language Support

Language support includes translated manuals, error messages, and commands. Sometimes number formats and dates are different in different languages. Sixteen-bit characters can be stored where appropriate.

5.7.13 Precompiler Support

Precompilers allow the embedding of SQL statements in Third Generation Language (3GL) programs. These programs are passed through a pre-compiler to produce pure 3GL code with low-level function calls replacing the SQL statements. Precompilers should also provide both syntactic and semantic checking for increased programmer productivity. Precompiler support is a key component of the ANSI SQL standard as well as the Federal Government's NIST SQL compliance test.

Precompilers permit a programmer to write standard code that can be re-precompiled and re-compiled and then will run against other database engines. If programmers write code to their product's function call interface, it will be difficult for them to change engines.

5.7.14 Web Support

New technology would indicate that the RDBMS tool should be able to reside on a web site on the Internet to make it accessible for all users, versus loading the RDBMS tool on a server or on individual PCs.

5.7.15 XML Support

Extensible Markup Language (XML) is a flexible solution for online document publishing. It has been designed for easy implementation and for interoperability with both SGML and HTML. Vendors should be asked if they are aware of XML technology, are developing an XML interface, or planning a release that includes XML technology.

5.8 Centralized Meta Data Repository Inter-operability (CMDR)

The SCIT initiative is intently focusing on a capability to develop and manage meta-data for all enterprise databases supporting the three agencies. This requirement is that the product line suite contains a totally integrated Meta Data Repository and administrative control feature to perform bi-directional real time export of system metadata from disparate data base definition languages, DDLs, and post updates the the centralized model.

Integrated reporting tools including graphical display of the system models will be considere as a plus for the selection of the DBMS suite.

5.9 Enterprise 7x24 Operations and System Administration

OLTP systems may have near-continuous up-time requirements. The following requirements for on-line backup, on-line reconfiguration, disk mirroring and other high availability features should be added.

The warehouse DBMS should provide automatic system recovery. Database failures are usually grouped into several categories. Instance failure occurs when the machine running the database server crashes, or software fails, or an uninformed operator damages the server, with no losses on the database disks. Media failure occurs when a disk containing database information fails.

Database recovery in case of instance failure should be performed automatically by simply restarting the database instance. The engine should roll back any transactions that were pending but not committed at the time of failure, and ensure the integrity of all committed transactions. The time to recover from instance failure should be configurable by the DBA.

Recovery from media failure should be automatic, semi-automatic, or manual. In all cases, recovery from media failure requires that a backup of the lost files at some point in time is available, along with all database redo files since that point in time.

5.10 Middleware

5.10.1 Middleware Embedded Support

Future component middleware will be embedded into other technologies. The vendor should have various middleware products that will complement an existing product line for an enterprise solution. How does the vendor define middleware and what are the various components for the product line?

Middleware requirements that become necessary parts of a database implementation may be inadequately addressed in Service Center environment configurations. An example of middleware is ESRI's Spatial Database Engine (SDE) product. These middleware products can add significant cost as well as functionality to a solution. Middleware can also add complexity in the administration, refresh, training, and implementation cycles.

5.11 Cost

5.11.1 Cost To Procure

This is an enterprise procurement for three agencies. Therefore, the answers to the following questions can affect planning and pricing.

- What are the vendor's various enterprise-priced licenses available?
- What are the Government schedules available?
- When is the vendor's next major release and how does this affect pricing?

5.11.2 Cost To Maintain/Upgrades

After the products are procured, maintenance costs occur during the lifecycle of those products. These costs could ultimately be more than the original procurement costs. The answers to the following questions will help determine those costs:

- Are there different levels of maintenance agreements?
- What do they cover?
- What are the costs?
- At what point after product installation are they required?
- Will they include any future upgrades at no cost?
- What is the policy on upgrades?
- How often are the upgrades delivered?
- At what point are the upgrades no longer free?

5.11.3 Cost for Training

Training is an important issue, since most staff members will not have the necessary knowledge to use these tools. The answers to the following questions will determine the extent of that training:

- What training will be required for personnel to use the product proficiently?
- Who is available to provide the training?
- What is the cost of training per student?
- What is the length of training classes?
- Where is the training offered?

5.11.4 Cost To Implement

Once RDBMS tools that are being used are identified, it is important to determine how many staff members use the tools. The answers to the following questions will determine the extent that members will use the tools:

- What percentage of personnel use the tool?
- Can the personnel take any training in the future?
- Can the trained personnel become consultants on using the tools for different projects?

5.12 Support – Requirements for DBMS

5.12.1 Technical Support

Technical support is essential in the use of any software and hardware. The answers to the following questions will determine the extent of vendor supplied support:

- What hours is the vendor available for technical support?

- What are the different levels of support in emergency situations?
- What is the turnaround time to the users when help is requested?

5.12.2 Vendor Future – Requirements for DBMS

5.12.3 Current Market Assessment (Financial)

The answers to the following questions will determine the whether the vendor will be around to support the tools:

- What is the financial standing of the vendor's company?
- Does the vendor have a long-term product support plan in place?
- Who are the vendor's clients?
- Does the vendor have any clients with similar needs and what are the clients' satisfaction levels with that product?

The team will use the Gartner Group as the source of this information.

5.13 Interoperability – Requirements for DBMS

5.13.1 Access to Open Protocols

The DBMS should use common communication protocol TCP/IP. The DBMS should not force the use of proprietary solutions such as Netware/SNA gateway. It should also allow means so a user can join tables from two vendor database tables. A means for providing for the built in import/export capability without writing SQL scripts should be included.

It should support client, midrange and mainframe portability and inter-operability of data and application procedures. Row and element multi-tier replication

5.14 New Technology Criteria – Requirements for DBMS

5.14.1 Web Server

The server should have the following abilities:

- Access the data warehouse using browser-based technology without losing any of the functionality and user interface of the client-server environment.
- Configure different levels of functionality by defining user and/or group privileges.
- Deliver a WEB interface with zero-administration to end users

5.14.2 Query & Reporting

The following abilities should also be incorporated into the enterprise:

- Automatically run scheduled queries and distribute reports to WEB servers, FTP servers, via E-mail, and directly to LAN printers.
- Schedule to run standard pre-defined queries and reports every time a data mart or data warehouse is updated, and distribute the queries and reports to WEB servers, FTP servers, via E-mail, and directly to LAN printers.
- Schedule using a point and click interface with all scheduling and distribution options using pull down menus.

- Complex querying and analysis using: an intuitive point and click, and drag and drop. User interface that requires minimal training.
- Satisfy the query and reporting needs of varying type of users from novice to the power user using the same user interface.
- Produce data cubes for desktop analysis with the following functionality: pivoting, drilling across, down, and up, ad hoc data sorts, ad hoc custom groupings, local calculations, weighted averages, and color graph and charts.
- Perform complex analysis such as mathematical, time series, and if-then-else functions on aggregate data as well as detailed data.
- Link data ad hoc or predefined from multiple sources into a single document for reporting and analysis, with the ability to automatically schedule these events and distribute reports to WEB servers, FTP servers, via E-mail, and directly to LAN printers.
- Link data, ad hoc, from external sources such as spreadsheets or flat files.
- Export query results to other products like Lotus or Excel. The ability to customize and reuse SQL that has been automatically generated.
- Share and reuse standard queries across the organization.
- Track or audit who is running what queries and how often.

5.14.3 Meta Data Management

The following Meta Data Management capabilities should also be incorporated into the enterprise:

- Automatically update the latest version of an object stored in the repository to all occurrences of that object in the data warehouse the next time the files in the data warehouse are accessed.
- Incorporate any metadata into the data warehouse, such as database specific information, metadata generated by third party tools, or a custom Data Dictionary inside the data warehouse.

5.15 Section 7 – Mobile Computing (MC) Requirements

Disconnected mobile computing applications are key to unlocking technical assistance and providing quality information products to service USDA customers. The DBMS is needed to build and deliver USDA business applications requiring ease of use, application interoperability, scalability and reliability to support disconnected mobile computing applications.

It should be understood that the basic features that apply to the selection of the DBMS should also apply to the Mobile Computing environment.

5.15.1 DBMS Support of Mobile Windows NT Workstation Applications

The DBMS must support mobile computing and other lines of business applications for USDA. We believe important areas of leadership and innovation in the DBMS will require the following:

5.15.2 MC: Enterprise Impact

Capability to scale from the laptop to the enterprise using the same code base, offering 100% code compatibility - resulting in 100% application compatibility from a laptop to enterprise SMP database servers.

5.15.3 MC: Systems Integration

Support a wide array of merge replication options for any database – enterprise to disconnected mobile computer.; Tight integration with Service Center CCE application platform configurations including Windows NT Server, Microsoft Office and the BackOffice family.

5.15.4 MC: Performance

High-performance access to a variety of relational database and non-database information sources.

5.15.5 MC: Specialized Requirements

USDA will want to protect investments as they scale database applications down to laptops and out to field offices. To accomplish this, USDA will need database engine technology that scales from a mobile laptop computer running Windows NT to enterprise-level symmetric multiprocessor clusters.

USDA needs a DBMS designed for the growing needs of the mobile computing requirements, with features like low memory footprint, automatic tuning and multi-site replication. The DBMS must also support workstation applications that reach data warehousing systems, thus requiring scalability features like dynamic row-level locking, parallel query, distributed query and very large database (VLDB) support.

The DBMS for Windows NT Workstation should be a fully featured RDBMS targeted for workstation and mobile applications. Common source code for all platforms — from Windows NT workstations to clustered servers — resolves compatibility issues. Mobile clients should be fully supported with merge replication and conflict resolution.

5.15.6 MC: Replication Support for Disconnected Computing

The DBMS should enable development of distributed solutions, including a large variety of applications that require data replication. For mobile computing applications, the data replication model should build on the “*publish and subscribe*” metaphor and replication software interfaces which should be available for custom third-party applications. Distributed merge replication should allow sites to make autonomous changes to replicated data, and at a later time, merge changes made at all sites. Merge replication may not guarantee transactional consistency, but it will allow the greatest amount of site autonomy – which is critical for disconnected computing applications. Update replication, where data replicated by the DBMS can be modified at multi-sites, should support different solutions appropriate for different applications.

The DBMS should also include COM interfaces that open up the store-and-forward replication services. This allows heterogeneous data providers to use the DBMS replication infrastructure to publish their data. The DBMS should provide completely heterogeneous data-replication services.

The DBMS should also include enhancements for Internet replication. Anonymous subscriptions and built-in support for Internet distribution to simplify data replication to the Internet.

5.15.7 MC: Performance, Reliability, and Scalability

The relational engine improves reliability, security, and performance while scaling from mobile laptops to terabyte SMP systems.

5.15.8 MC: Replication

The DBMS should deliver a broad spectrum of innovative Replication technologies for building distributed business applications.

5.15.9 MC: Ease of Use.

Simplified user interface with wizards, improved monitoring, scripting, and troubleshooting.

5.15.10 MC: Heterogeneous Support.

Standard published APIs which support bi-directional replication with other data providers like Oracle, DB2, Sybase and Informix. Replication to non-relational data stores is also supported via third party solutions.

5.15.11 MC: Immediate Update.

Changes to a Subscriber’s data can be immediately propagated to the Publisher via a two-phase commit, and then to other Subscribers using Transactional or Snapshot replication.

5.15.12 MC: Internet Support.

Anonymous pull subscriptions allows servers on the Internet to subscribe to publications without having to register with the publisher. This model allows large numbers of servers to participate in the DBMS replication.

5.15.13 MC: Merge Replication.

Merge is a new replication model in which users work freely and independently. At a later time the work is combined into a single uniform result. This model is ideal for offline or disconnected applications.

Methods will provide resolve merge conflicts via priority-based resolution. A standard interface is needed to provide support for business rule reconciliation.

5.15.14 MC: Multi-Site Update.

Allow updates on multiple copies of the same data at different locations.

5.15.15 MC: Scalability.

Support replication to hundreds of servers and thousands of users through a streamlined architecture that reduces contention on replication tables.

5.15.16 MC: Desktop and Mobile Systems

The DBMS should scale downward to provide a fully featured RDBMS targeted for workstation and mobile applications. Common source code for all platforms from Windows NT systems to clustered systems resolves compatibility issues. Mobile clients are fully supported with merge replication and conflict resolution.

5.15.17 MC: Automatic Tuning.

On-demand memory and disk tuning, dynamic locking and minimal tuning parameters to provide simplified administration.

5.15.18 MC: Code Compatibility.

100% code compatibility to provide the ability to use the same source code across all platforms.

5.15.19 MC: Embedded Version.

Independent software vendors (building sales force automation software, for example) can easily license the lightweight, full-featured, low-cost database engine and core components.

5.15.20 MC: Integration With Microsoft Access.

Provide necessary integration with Microsoft Access to allow simplified development, prototyping, and upsizing for Access applications.

5.15.21 MC: Mobile Clients and Replication.

Merge replication simplifies the development of applications for mobile clients.

5.15.22 MC: Universal Data Access.

Universal Data Access should be the enabling high-performance access to a variety of information sources: OLE DB and ADO that build on the wide support for ODBC.

5.15.23 MC: Integration with Microsoft Office 2000

DBMS will integrate with MSDE which is an enabling technology that provides local data storage for the DBMS compatible applications. It is an alternative to the Jet database engine used by Access 2000.

5.15.23.1 MC: Connectivity.

Connectivity to the DBMS and other databases is improved with a client/server layer that uses OLE DB and ADO.

5.15.23.2 MC: Developer Tools.

DBMS provides tight integration and facilitates rapid development for custom Office-based solutions based on the personnel productivity software field on CCE client computers.

5.16 CASE Tools Interoperability

The product shall be compatible with agency specific existing CASE tools including: PowerDesigner and CoolGen. The dbms product/vendor suite shall demonstrate full interoperability and data management for the stated CASE tools, and for any other proposed by the vendor.

5.17 Questions for Vendors – Requirement for DBMS

The following question should be answered by prospective vendors:

The following question should be answered by prospective vendors:

- Describe how and when are the system components integrated.
- Describe the joint development of DBMS products between your firm and other firms.
- Describe how are the product releases synchronized.
- Define who is responsible for this function.
- What are the Vendor's recommendations for resolving connectivity and performance tuning problems.
- Does the Vendor's product support multiple DBMS and platforms for performance?

6 Section 6—Data Warehouse (DW) and Data Mart Requirements

The evaluation strategy (Section 4.1) is to examine each category of application and explain the technical characteristics the proposed product must have to be successful in the category. The technical evaluation matrix (Section 4.2) will list individual requirements that support each characteristic. In this way, each requirement will be driven by an actual application environment and can be traced back to the agency objective that generated it. Characteristics and requirements are grouped into four large areas:

- **Architecture and Performance**
This area identifies the architecture of the product and its optimization on appropriate hardware configurations. It assesses hardware requirements imposed by the data management tool and assesses scalability, optimization and concurrent user access.
- **Administration**
This area examines administration characteristics of the product including the ease of installation, configuration and maintenance. In addition, it examines security and integrity issues.
- **Functionality and Extensibility**
This area determines the suitability of the product to meet current and future USDA objectives. It examines the ease with which basic and advanced processing may be accomplished how the product will fit into the USDA environment. Real-time and batch interfaces are considered including replication methods, gateways to access foreign data repositories and middleware solutions. User access via an intranet and client/server tools is also examined.
- **Highly Complex Documents**

Refer to Section 4.1.3 of this document.

This section is intended to act as supplement to the SCIT DBMS Data Management Tools Selection Strategy criteria contained in Section 5 above. The express purpose here is to articulate the baseline requirements derived for the Data Warehouse Initiatives, including pilot projects and benchmark testing. A Data Warehouse Technical team has been established for the express purpose to address this subject issue and is to be the point of contact for coordination by the CCE for test evaluations and selections.

6.1 DW: Architecture

Data Marts will serve as the mid-tier in the USDA architecture. They will be hosted on multi-processor (SMP) and single processor hardware. Marts must provide efficient, high performance access in both configurations and should support a variety of UNIX and NT platforms. Data operations (load/unload, query, update, index building, etc.) should be performed in parallel when multiple processors are available.

6.2 DW: Specialized Schemas

To support decision support operations, it must be possible to configure the DBMS using specialized schemas including Star or Snowflake data structures. It should be possible to add business intelligence to the DBMS indexing capability.

6.3 DW: Operational Environment

This section outlines the future environment that the data management tools must support. They must support collecting and disseminating of information in a multi-tiered environment. The Data Management Tools TWG recognizes the requirement of Service Centers to collect and manage local information that is not shared outside their internal organizations.

6.4 DW: Operates within the Common Computing Environment (CCE).

Operates within a fully integrated/scalable open architecture; i.e. Mobile Workstation, NT Workstations, Midrange Server,...Mainframe. Therefore the dbms must support full heterogeneous platform independence across the enterprise. The same DBMS supports an interagency structure that builds on existing successful efforts to construct and operate data repositories which increases manageability, reliability and quality of data over the entire enterprise

As the central clearinghouse of the USDA architecture, the Data Warehouse Data Base must be open and interoperable with operational systems throughout the agency. It must support a variety of extraction, transformation and loading utilities through standards-based interfaces

6.5 DW: Define an Enterprise Architecture that evolves and scales

Begin with prototypes that integrate data from a few related legacy systems from more than one agency component, and transition toward the Conceptual Enterprise Architecture and National deployment.

Establish a single integrated information management architecture, reduce redundant data, redundant processes, enhance transaction processing, data mining, and data analysis capability through a single, simple interface.

6.6 DW: ODBC Support

The Vendor should provide (at no charge) an installable, standalone ODBC driver. Standalone infers that the ODBC driver should not be integrated with any of the Vendor's other products.

6.7 DW: Security

Provide security to protect access to the data and protect the integrity of the data (similar to C2 level of trust) and provide integrated information to users with a need-to-know.

6.8 DW: UNIX Compatibility.

Unix, as a minimum, should be the required platform.

6.9 DW: Compatible Data Recovery.

Must utilize RAID or compatible data recovery techniques

6.10 DW: Multi-tasking.

Simultaneously multi-tasking across multiple processes

6.11 DW: Infrastructure (Telecommunications) Compatibility

- Data Warehouse/ Mart platform – UNIX
- Cabling for DW/DM infrastructure – 100base-T
- Transaction collection and distribution points – On-Line 56k baud
- Network performance projections, staff skills, and business communication capabilities will determine the locations of centralized and decentralized data servers.

6.12 DW: Business Objectives

The Enterprise Architecture implementation will accomplish these business objectives:

- Create a program leadership environment that includes the leadership of the business groups with ownership of the legacy systems to ensure that there is both understanding and agreement that the Enterprise Architecture will meet their current and future needs.
- Facilitate timely response to changes in USDA policy, regulation, and law.
- Quickly establish a cost-effective process to intelligently use information now in stovepipe systems.
- Provide a resource for USDA partner agencies to increase productivity to accomplish the mission and reduce maintenance costs.
- Enhance interoperability of user organizations by providing access to integrated data, minimize unnecessary duplication of effort; and capitalize on agency successes.
- Meet or exceed the capabilities of the legacy systems in use at the time of implementation.
- Meet the demand for interoperability and requirements for superior performance and efficiency.
- Provide an audit capability to track actions taken on a record and associate that action with the initiator.
- Exploit the use of COTS/GOTS.

6.13 DW: Architecture

A Data Warehouse complex with multiple Data Marts requires more functions than are represented by the hardware/software/data of the repositories. Specific requirements for Data Warehousing are discussed within this section.

6.13.1 DW: Multiple Staging Areas with Synchronization

Ability to have multiple staging areas that can be replicated to other enterprise servers. Synchronization of data in multiple staging areas/databases

6.13.2 DW: Data Administration.

Online Data Administration and Management System functions supporting loading from multiple sources including DBMS and warehouse to warehouse, internet interface reporting

6.13.3 DW: Extraction, Transformation, and Translation

NOTE: A Data Warehousing Document for ETT is to be developed and will delineate standards and system requirements. Once developed, the requirements contained therein shall be used for application to the Data Warehouse DBMS selection.

6.14 DW: Data Delivery

The data delivery function is the dynamic mechanism that builds and manages Data Marts as well as many smaller data deliverables from the Data Warehouse's large store of information. The basic processes of data delivery is:

- Filter for content,
- format/design for standard structure,
- deliver data to the DM in accordance with requirements (content and time period) of the user group of the DM,
- update the DM as the DW is refreshed

The data delivery function provides facilities to support approved requests, schedules resources with the priority of approved requests, tracks and monitors requests in progress, and provides requirements for and use of templates in automation of service requests.

The data delivery function must support a variety of output formats and analytical platforms required by the user's environments, such as comma delimited, multidimensional, relational, etc. The data delivery function delivers data in a form to make it easy for the user to perform their task. Data delivery will either be scheduled or demanded by a high-priority query.

6.15 DW: High Speed and Volume Loader with Disparate Sources

As the middle tier, the Data Marts will be regularly populated through load utilities. The Data Mart DBMS should be evaluated for the speed, high volume and complete interoperability with other DBMSs including: Oracle, DB2, SYBASE, INFORMIX, CA Open Ingres, SQL Server, etc.

The DW is the source of data population for Data Marts. Few analytical research activities will be performed using the DW itself; the majority of analyses will use a Data Mart. However, if the analysis requires data outside the scope of the Data Mart, the DW should provide the resource for analysis

6.16 DW: OLAP Tool and Reporting Extensions

In addition, support of OLAP (Decision Support) Tools is evaluated. Features which support a wide-range of OLAP tools, both client-server or over the Web will be a requirement.

Although analysis will usually be performed against the data mart specific to a business process, reports may be run directly against the data warehousing. For instance, Data mining to discover relationships and/or operational reports on historical, normalized, non-aggregated data will be supported at the warehouse level.

6.17 DW: Loader Customizations

Features which extend the analytic capabilities of the DBMS will be evaluated. These include extended business rules, custom analytic functions, custom indexing and custom sorting options.

Within the USDA, data from multiple source legacy systems will be transformed and cleansed, and, in nearly real-time, feeds the USDA Data Warehouse. In turn, the USDA Data Warehouse will serve as a staging area to normalize and integrate data from disparate operation systems before feeding sub-sets of that data into specialized data marts. It is in the consolidated Data Warehouse that new and meaningful relationships will be built out of previously isolated applications. Data integrity will improve by standardizing on the metadata model of the warehouse. Total cost of ownership will improve through central administration of the data and the interfaces.

6.18 DW: Data Warehouse to GIS Extensibility

As subsets of the Data Warehouse, some USDA data marts may handle a combination of geo-spatial data and business data. The architecture of these geo-spatial marts should support OLAP tools through standard interfaces (i.e. ODBC.)

The warehouse must be logically integrated and may be physically distributed according to the needs of the business.

The function of the DW is to provide a logically centralized and accessible repository where captured, cleansed, integrated and authoritative data are stored. The DW is a resource for analytical activities that supports decision support, executive information, and strategic planning. It also is a resource of integrated, standard and authoritative data, or data for analytical or operational activities. Users view the DW as a seamless, consistent, and accurate view of enterprise data across functional boundaries.

Data Marts (DM) can be managed by departmental or functional Information Technology (IT) resources. Operational Data Feeds (ODFs) or special data feeds are designed not to burden the user's system resources. As the Analytical Services evolves to the Enterprise Architecture, it may be instituted as a special database for the DM and replaced/refreshed as the data repository matures. Implementation is dependent on the analytical tool(s) employed by the functional organization. Because of the variety of user needs, queries are unpredictable and may be complex, and may return relatively small sets of data.

6.19 DW: Multi-tasking and Parallel Processing

Does the product support parallel processing alternatives of multi-CPU parallel processing hardware to optimize resource management?

The product must meet high demand load criteria in minimal load times to be accessed and must allow concurrent batch load and user online inquiry.

6.20 DW: Transaction Processing Council Benchmarks

The product to qualify should have been benchmarked on the TPC-C (Data Warehousing) tests. The scores should be viable information for speed and capacity analysis.

Have TPC build a matrix of transaction volume demand, database size, hardware specs, software, number of users and performance scores to create a map of preferences that will depend on the enterprise growth parameters.

6.21 DW: Specialized Requirements

6.21.1 DW: Very Large Databases (VLDB)

The USDA Data Warehouse will likely be the largest single component of the agency architecture. It must provide high performance access to Very Large Databases (VLDB) and linear scalability to grow as years of historical data are retained.

6.21.2 DW: Symmetrical Multiprocessing (SMP)

At a minimum, the Data Warehouse DBMS architecture must be optimized for large symmetrical multiprocessing (SMP) hardware systems. At a minimum, it should efficiently support Massively Parallel Processing (MPP) systems or loosely coupled systems. Trade-offs between different hardware architectures should be discussed.

6.21.3 DW: Splitting a table(s) across many physical devices

The DBMS should have mechanisms for splitting a table across many physical devices. It should be possible to partition data in round-robin inserts (maximum performance on randomly occurring data) or based on value or expression (to separate data by month, for instance.) The optimizer should consider the partitioning of data when developing a query plan.

6.21.4 DW: Load/unload, query, update, index building within multiple processors

To support performance and size requirements, all data operations (load/unload, query, update, index building, etc.) should be performed in parallel when multiple processors are available. The use of parallelism should not limit DBMS functionality in any way. It should be possible to use multi-statement transactions, integrity constraints, serializable transactions and row-level locking on parallel operations. It should be possible to perform all data operations in parallel operations when tables or indexes are split across multiple disk drives.

6.21.5 DW: Asynchronous I/O

The DBMS should support asynchronous I/O where available from the operating system and should have a strategy for dealing with asynchronous I/O where is not available.

6.21.6 DW: Clustered Indexes

The DBMS should support clustered indexes (sorted on a cluster key or pre-sorted) to speed data retrieval on ordered operations.

6.21.7 DW: Query Optimization

The DBMS should support cost and statistics-based **query optimization**. Because of the large size of warehouse tables, it should be possible to gather statistics based on sampling. The optimizer should consider the partitioning of data when developing a query plan.

6.21.8 DW: Terabyte commercial installations

The data warehouse DBMS should have demonstrable references for **multi-terabyte commercial installations** and should have been tested against industry standard decision support benchmarks.

6.21.9 DW: SQL92 for data definition and data manipulation

SQL92 for data definition and data manipulation (To ensure the widest compatibility, the product should comply with FIPS 127-1.) This emphasizes the use of standard non-proprietary SQL.

6.21.10 DW: SQL implementation (null or not)

To support consolidation and integrity checking of data, Warehouse DBMS should support Declarative Referential Integrity, default values and domains. It should support null values in tables. “The SQL implementation should provide a function that will determine whether the value in the column is null or not. This feature is useful when performing arithmetic operations and outer joins. The database should also correctly evaluate to false null = anything. “

6.21.11 DW: Outer Joins

The Warehouse DBMS should support outer joins. “Outer join lets users perform a join where they want all the rows from one table, even if no matching rows exist in the second table. This feature is very useful for modeling real world problems”.

6.21.12 DW: SQL92 data types (business information)

In addition to SQL92 data types (business information), the USDA Data Warehouse should be capable of maintaining complex data—at a minimum geo-spatial data. The ability to build a spatially aware data warehouse will allow policy makers and managers to capture all of the important dimensions of agency-wide information. For instance, inclusion of spatial functions in the warehouse will permit analysis of data by time, by amount, and by spatial relationships such as “nearness”.

6.21.13 DW: Spatial Data

Technology, along with user requirements, will expand and change through the life of the Data Warehouse. In addition to spatial data (which is a current USDA requirement), the Gartner Group predicts an increase in the use of text and image data in new information systems. This is clearly the case at USDA, where HTML pages, XML documents and DBMS management of digital ortho images are anticipated.

6.21.14 DW: Extensible for new types of data and standards

To support future requirements, the Data Warehouse DBMS should be extensible to easily support new types of data and to support new industry standards such as SQL93.

6.21.14.1 DW: Variety of options/tools for performance optimization

Due to its large size, knowledgeable performance tuning and timely administration of the data warehouse will be critical. The Data Warehouse DBMS must offer a variety of options for performance optimization, and must be easy to configure and administer. The DBMS should have tools to monitor system performance and diagnose problems. The tools should monitor individual SQL statements and overall system performance.

6.21.15 DW: Variety of Backups

A variety of backup options, including third party parallel back-up utilities should be supported. Backup should be possible without taking the database off-line. Multi incremental backups should be possible, whereby only the changed blocks are backed up. This should reduce the time required to do a backup and allow one or more tablespace recovery while the remaining database is up and running..

6.21.16 DW: Single warehouse for spatial data and future use

Business, spatial data and future complex data should be able to be administered as a single warehouse. The inclusion on extended datatypes should not compromise the performance, ease-of-use or integrity of the warehouse.

6.21.17 DW: Backup Warehouses

Hot shift to a backup data warehouse if the primary warehouse goes down. (i.e. 7x24 availability)

6.21.18 DW: Efficient loading of the Data Warehouse into Data Marts

Efficient loading of the Data Warehouse from agency-wide systems and efficient extraction from the warehouse to into USDA data marts will be critical to the smooth operation of the USDA.

6.21.19 DW: Parameters supporting the migration of data from legacy systems

The Warehouse DBMS should support a variety of extract/transform/load utilities. The data warehouse DBMS should be evaluated for the ease and speed of data extraction and loading and support of “encapsulation techniques” such as the CORBA and COM data models.

Because Data Marts will be smaller, more numerous and more localized than the USDA Data Warehouse. Accordingly, features such as parallel backup will be weighted less and ease of installation and administration will be weighted more for the Data Mart DBMS than for the Data Warehouse DBMS. Although it would be convenient for Data Marts to have the same administration tools and architecture as the Data Warehouse DBMS, more consideration will be given to choosing the right Data Mart DBMS for the right analytic application. Therefore DBMS which are specially designed for Data Marts will be considered.

6.21.20 DW: Large Volume Data Transfers

Data Marts will load data primarily from the Data Warehouse. Accordingly, the ability to extract, transform and load from legacy systems will be weighted less in the Data Warehouse DBMS. On the other hand, Data Marts must be able to accept large data transfers from the Data Warehouse DBMS using generally smaller hardware. Therefore, parallel data loads from the warehouse will be weighted more.

6.22 DW: Decision Support Systems

DSSs, applications running at the agency level, provide input into the decision-making process of an organization. Traditionally, these will consist of a data mart focussed on a single functional area plus Decision Support Tools.

6.22.1 DW: Highly Complex Data/Documents

A number of architectures can be used to integrate complex documents into a DBMS. For instance, documents can be indexed using a separate full-text retrieval engine and stored as binary large objects in the DBMS. Alternatively, the DBMS engine can be extended to understand and index documents stored as complex objects.

In the USDA environment, query performance over documents, especially those that cross-reference other documents or contain many components is the most important performance requirement. It is anticipated that documents will be linked by attribute information (i.e. documents relating to a specific directive), component (i.e. heading, abstract) as well as by textual meaning. This will require complex ad-hoc searches spanning several index types (i.e. full text and relational.)

Architectures should be evaluated on the basis of how well DBMS and text data and indexes are integrated, the range of storage choices for documents (file system or DBMS), and the way complex queries are processed. Integrity issues when multiple servers are updated in a single transactions should be considered.

6.22.2 DW: Functionality and Extensibility

To meet future requirements for complex document support, the DBMS must be able to define the structure of complex documents to the DBMS. The structure must maintain the identity of the overall document while providing access and cross-referencing of individual components. Documents and components should be searchable by attribute (date, author, etc.), by word, or by context (fuzzy search.)

Future requirements may tie document formats to document type definitions, such as those provided in XML. It should be possible for the DBMS to “parse” documents to understand their structure and content. The ability to modify or extend this capability in the future is critical.

6.22.3 DW: WEB Enabled Document Transfer and Reports

Most document access will be from Web applications. The ease of building an application that inserts and queries documents should be considered. Performing document data manipulation through SQL (as opposed to 3GL API's) is strongly preferred.

6.22.4 DW: Application Development Tools (Inclusion / Interfaces)

More than any other category of application, OLTP systems require excellent support of Web and client-server application development tools. Support for popular integrated development environments (Such As: Visual Basic, Cool Gen, PowerBuilder, etc) and third Generation languages (C, C++) should be extensive and standards-based.

Since there seems to be a strong industry trend toward Java, the DBMS should support JDBC and n-tier Java application servers. The OLTP DBMS should offer a variety of Web options including support of third party tools such as Cold Fusion and Net Dynamics.

6.23 DW: Query Performance

6.23.1 DW: Parallel index scan and parallel execution plans?

Provides parallel index scan and parallel execution for all DML operations such as: update, delete, insert and select. This should allow users to take advantage of parallel insert, update, and delete on partitioned tables.

6.23.2 DW: Query rewrite facility

Enables query rewrite facility whereby poorly written SQL statements that specify a star join are rewritten and optimized for performance..

6.23.3 DW: Thread Performance

Performance threads along the OLTP and OLAP paths need to be defined based on user needs. Performance threads also need to be defined in the internal processes of extraction, cleansing, integration, transformation, transport/load from the Operational segment to the Analytical segment warehouse and again to the mart.

7 Section 7 - Message Queuing and Middleware Support

Messaging and middleware will be used to implement BPR applications, these products will be evaluated on n performance and how well the middleware supports the vendor's product line. DBMS vendors should discuss development strategies, messaging and middleware products that will complement an existing product line for an enterprise solution.

To the greatest extent possible, spatial data should be indexed by the DBMS, be searchable through SQL, and be available to analysis tools and Web applications through standard interfaces such as ODBC and JDBC. It should be possible to query on any combination of business data and spatial data. The DBMS should automatically optimize the performance of such queries based on the data distribution of both business and GIS data.

7.1 Image Management

The GIS DBMS should be capable of managing Digital Ortho Quad (DOQ's.). It should be able to tile DOQ's to present a seamless appearance to the user and extract partial images from DOQ's. It should be possible to describe coordinates of DOQ's in multiple-user specified coordinate system and datum (Latitude/Longitude versus UTM).

The GIS DBMS should be able to manage images in compressed formats including JPEG, PNG and MrSID.

Ideally, the GIS DBMS should be able to perform Boolean algebra on raster graphics, such as classified imagery, inside the DBMS, without resorting to an external GIS engine.

7.2 Vector Data Management

The GIS DBMS should provide the ability to store, retrieve, query, and update Vector data. Updates should include transaction semantics for feature locking. Ideally, vector updates should be versions which will maintain vector data history and allow back tracing a series of operations. Assuming history had been maintained, it should be possible to reconstruct vectors based on the history.

Ideally, the DBMS should provide the ability to perform Spatial operation on vector data inside the DBMS, without resorting to an external GIS engine and should support data in multiple user-specified coordinate system and datum (Latitude/Longitude versus UTM).

8 Section 8 – Geographical Information Systems (GIS)

The GIS system should use a commercial DBMS as the repository for spatial objects. The DBMS should support a variety of GIS systems. It should be capable of storing spatial data objects (location points, flood plains, etc.) together with the business data they describe. It should support the storage and manipulation of image formats (Digital Ortho Quads) both inside the database and on external devices, such as CD's.

Like complex document data, a number of architectures can be used to integrate spatial data into a DBMS. Also, like document data, speed of querying ad-hoc combinations of attribute data with complex spatial data is the primary requirement of the GIS DBMS. The DBMS should be capable of determining the fastest path to data, regardless of how the data is distributed or queried (attributes first, spatial first, attributes primarily, spatial primarily, etc.). It should be possible to view the query plan for the combined DBMS/spatial query and to tune it if necessary. Parallel operations on spatial data should not be sacrificed

If the DBMS requires tiles to support GIS capabilities, it should provide the ability to combine tiles to present a seamless appearance to the user. Ideally, the DBMS should not require tiles to support GIS, but should handle data as single, contiguous layers which may be selected from to support any subset of data, large or small.

8.1 Management of Imagery (DOQ)

The product should provide the following DOQ functions:

- Efficiently manage imagery,
- Tile data to present a seamless appearance to the user,
- Query a DOQ that provides for partial extraction of a subset,
- Store data in multiple-user specified coordinate system and datum (Latitude/Longitude versus UTM),
- Perform Boolean algebra on raster graphics, without resorting to the GIS engine.

8.2 Ability To Manage Compressed Data

The vendor's product should provide the ability to manage compressed data, specifically MrSID, PNG and JPEG and similar forms of compression. It should perform this transformation seamlessly to the end user and have real time responsiveness.

8.3 Vector Graphics

The vendor's product should:

8.3.1 Boolean Algebra.

1. Perform Boolean algebra on vector graphics, without resorting to a GIS engine, Ability to store, retrieve, query, and update (transactions with feature locking) Vector data,

8.3.2 Multiple User-Specified Coordinate System .

Ability to store data in Multiple User-Specified Coordinate System and datum (Latitude/Longitude versus UTM), and have the ability to transform data between coordinate systems, e.g. State Plane, UTM (NAD83, NAD27).

8.3.3 Ability to Tile GIS.

Ability to tile data to present a seamless appearance to the user.

8.3.4 DBMS to maintain vector data history

Ability of the DBMS to maintain vector data history (back tracing a series of operations) and to reconstruct vector graphics based on that history.

8.4 Support for GIS Spatial Query Operators

The vendor should support the full set of Geospatial Query Operators, such as:

- (Not) touching
- (Not) disjoint
- (Not) crossing
- (Not) overlapping
- (Not) within
- (Not) contains
- (Not) equals.

Operations can be point-polygon, point-point, point-line, line-polygon, line-line, and polygon-polygon.

To provide maximum interoperability, the GIS DBMS should provide support for Open GIS Consortium Interoperability Specifications

9 Section 9 – Complex Data

Documents in the DBMS should be easy to insert, index and administer. The ease of installing and configuring the database that combines business data with complex documents should be evaluated. Because of their size, a variety of backup options for the documents should exist (backup always, never, when changed, etc.)

Requirements for Complex data have been defined in detail in Sections 5 and 6 above.

10 Section 10—Conclusion

The data management tools TWG met over a period of 4 months. During this time, there were numerous discussions with database professionals at various levels of the three agencies, industry consultants (Gartner Group), and workgroup sessions. The following are some considerations that highlight elements that the members found to be significant in a decision-making process:

1. There may not be a single enterprise-wide database solution.
2. Market share and future viability of the product or product combinations.
3. Multi-platform availability and scalability. NT and UNIX will coexist in the enterprise with Windows and MVS on small business platforms as well as enterprise mainframes. Look for interoperability of data and applications among systems rather than hardware and software solutions.
4. Interoperability solutions, implementations, and commitment are of significant importance.
5. Support is needed for extended data types such as XML.
6. Performance in the DBMS, while important, is not as significant a factor as the administrative overhead of the product and the interoperability issue from the perspective of maintenance of applications.
7. Performance in the loading and query area of the Data Warehousing DBMS is a significant factor, and should be considered before the administrative overhead of the product and the interoperability issue from the perspective of maintenance of applications.
8. Because integration of GIS into a DBMS environment is in its infancy, providing GIS data in a DBMS that resides at the Service Center (rather than at a National or Regional Server) may be cost prohibitive. In the short term, a heterogeneous GIS/database solution may be appropriate.
9. For example, ESRI's SDE is designed and priced for large databases hosted by a small number of servers rather than small databases hosted by a large number of servers. The cost of putting SDE in every Service Center could run \$30 to 50 million dollars. This cost does not include the DBMS or the GIS-related software. If this is the case, it might be necessary to manage GIS data like CLU at the state/national level with client access at the Service Center. Geospatial data that would need to reside at the Service Center might need to be kept in file systems such as shape files.
10. As multi-vendor interoperability specifications (such as those of the Open GIS Consortium) are implemented over the next 2 years, individual platform selections will become less relevant than the cross-platform availability of applications and data. This is true for hardware platforms, operating systems, office environment, database solutions, and GIS applications.
11. The implementation of interoperability specifications should play a major role in USDA data warehousing and dissemination strategies over the next 2 years.

LOG OF CHANGES :

June 1, 1999

Version DBMS3 is the newest File

1. Mobil Computing was moved to the DBMS Section 5.
2. Removed the 6.17 Operations test from the DW and added to the DBMS section.
3. Section 6.3 and 6.6 were combined to remove the duplication.
4. Section 6.19.1.6 modified to say “such as” in reference to the referenced tools
5. Changed 6.1.7 through 6.7.3 to level two paragraph.
6. DW 6.19.1.4 was moved to DBMS
7. DW 7 x 24 availability was moved to DBMS
8. DW 6.19.1.1 through 6.19.1.3 was renumbered
9. Changed the Ranking Table Titles to reflect the Platform(s)
10. In section 5, added performance metrics and the current categories of performance was retitled to Operations and Controls.